Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Large-Scale Algorithmic Synthetic Dataset for Pixel-Level Segmentation of Internal Pipeline Corrosion

Mohammad Hossein Babajani

moh.babajany@email.kntu.ac.ir

K.N. Toosi University of Technology

Ali Najafi Ardekany

K.N. Toosi University of Technology

Samir Neisy Minaei

K.N. Toosi University of Technology

Research Article

Keywords: Internal Pipeline Corrosion, Synthetic Dataset Generation, Pixel-Level Segmentation, Computer Vision, Semantic Segmentation, Deep Learning.

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11563524/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Large-Scale Algorithmic Synthetic Dataset for Pixel-Level Segmentation of Internal Pipeline Corrosion

Mohammad Hossein Babajani[1]*, Ali Najafi Ardekany[2], Samir Neisy Minaei[3]

[1]Master of Science Student, K.N. Toosi University of Technology, Tehran, moh.babajany@email.kntu.ac.ir

[2]Professor, K.N. Toosi University of Technology, Tehran, Najafi@kntu.ac.ir

[3]Doctor of Philosophy, K.N. Toosi University of Technology, Tehran, s.minaee@email.kntu.ac.ir

Abstract

Automated inspection of internal pipeline corrosion using computer vision requires diverse image datasets with accurate pixel-level labels. However, collecting and manually annotating real data is time-consuming, costly, and constrained by access and imaging conditions, which hinders the development of deep learning models. To address this limitation, an algorithmic, parametric framework was developed to generate synthetic images and corresponding masks. The framework incorporates cylindrical pipe geometry, a virtual camera, and illumination, using direct pixel-based mapping of an unwrapped two-dimensional surface without requiring an explicit three-dimensional model. Its methodological contribution is the integrated generation of images and masks from a shared spatial definition of corrosion and the preservation of their correspondence during geometric mapping. The framework also incorporates separation of the spatial pattern from its visual texture, sediment modeling, non-uniform illumination, and generation of multiple views from a common profile. Accordingly, the data were split at the profile level so that all views of each profile were assigned exclusively to the training, validation, or test subset. The final dataset contains 6,576 image–mask pairs at a resolution of 1024 × 1024 pixels and represents a quality-controlled sample of the generator's output space; this number does not represent its maximum generation capacity. Three segmentation architectures achieved Dice scores of approximately 0.973–0.974 on the held-out synthetic test set. The visual consistency of their predictions on real laboratory images was also assessed qualitatively, without fine-tuning. These results and observations support the utility of the dataset produced by the framework for developing and conducting controlled evaluations of corrosion segmentation models.

Keyword:

Internal Pipeline Corrosion, Synthetic Dataset Generation, Pixel-Level Segmentation, Computer Vision, Semantic Segmentation, Deep Learning.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

1 Introduction

Corrosion threatens the reliability of pipelines carrying oil, gas, and other fluids [1]. Because access for inspection is difficult, internal corrosion may remain undetected and lead to wall thinning and increased operational risk [2]. Its early detection and assessment are therefore important for predictive maintenance and pipeline integrity management [3].

Recent advances in imaging methods and artificial intelligence algorithms have enabled the development of automated systems for corrosion inspection and assessment [4]. Computer vision and deep learning methods have attracted attention for the automated detection and analysis of corrosion-damaged regions because they can extract complex patterns from images [5]. Deep learning models, particularly convolutional networks and newer segmentation architectures, have shown promise in identifying corroded regions and extracting spatial information from images [6]. However, detecting the presence of corrosion alone is insufficient for precise inspection; accurately locating damaged regions and delineating their boundaries is also important. Pixel-based semantic segmentation methods have therefore been developed as an effective approach to detailed corrosion image analysis [6].

The development of these models depends on sufficient and diverse training data [7], whereas acquiring suitable images and annotating corrosion at the pixel level is time-consuming and subject to practical constraints [6, 8]. The automated generation of diverse synthetic images and corresponding ground-truth masks offers a practical approach to addressing data limitations in industrial segmentation applications [6, 7].

Synthetic data enable control over sample generation, variation in imaging conditions, and the generation of corresponding labels. In industrial computer vision, SYNOSIS focuses on the controlled generation of surface-defect images [9], while SORDI.ai focuses on large-scale data generation [10].

Among synthetic datasets, VoroCrack3D illustrates the use of semi-synthetic data with precise annotations to develop computer vision models and highlights the importance of structured synthetic dataset design [11]. In corrosion research, synthetic data generation has also been investigated to address limited access to pipeline images [8] and to train models for other industrial applications [12].

K-Pipelines was introduced to provide synthetic data for pipeline corrosion classification. Its annotations are at the image level, and it does not provide pixel-level masks for corrosion segmentation [8]. Corrosion segmentation methods, meanwhile, have demonstrated the ability to delineate damaged regions accurately. However, some of these studies rely on available real data and face challenges such as limited sample diversity and the time required to produce precise masks [6, 13].

Studies of inverse perspective mapping and cylindrical surface unwrapping primarily correct the geometry of internal pipe images or reconstruct surface views [14, 15]. Siu et al. also used three-dimensional simulation to generate synthetic images of internal sewer-pipe defects for defect detection [16]. Among the selected studies, the integration of algorithmic generation of corrosion patterns and appearance, corresponding pixel-level masks, and direct mapping from a two-dimensional surface to an internal pipe view for corrosion segmentation has not been reported in the form used in the present study.

To address the need for internal pipe corrosion data with corresponding pixel-level masks, this study presents two complementary contributions: an algorithmic, parametric data generation framework and a specialized dataset produced using that framework. The framework separates the background texture, corrosion texture, and spatial damage pattern on a two-dimensional surface and directly computes the coordinates of each pixel in the cylindrical internal view on that surface. It therefore requires neither a three-dimensional model nor a geometric mesh. The image and initial mask are generated from a shared spatial

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

definition of corrosion and use the same coordinate field during cylindrical mapping; image appearance processing and mask corrections are distinct at other stages.

The generation pipeline incorporates variations in background texture and color, sediment at the circumferential seam of the unwrapped surface, non-uniform illumination, and terminal-region processing. Axial displacement of the virtual camera also produces multiple views from a common profile. The profile identifier records the relationship between views, and profile-level splitting prevents views of the same surface from appearing in different subsets.

The final dataset contains 6,576 image–mask pairs at a resolution of 1024 × 1024 pixels, selected from 7,464 initially generated pairs after 888 pairs were excluded during visual quality review. Its utility was assessed using three segmentation architectures under a common protocol. Predictions from models trained on synthetic data were also examined qualitatively on 116 images from ten physical pipe specimens, without retraining or fine-tuning. The generation framework and the structured dataset thus constitute two complementary contributions of this study.

Section 2 presents the generation method, Section 3 describes the dataset structure, Section 4 compares the study with related work, Section 5 evaluates dataset utility, and Section 6 discusses the findings, limitations, and future directions.

2 Algorithmic Synthetic Dataset Generation Framework

The proposed framework transforms an unwrapped two-dimensional representation of the internal pipe surface and its corresponding mask into a cylindrical internal view through direct pixel-coordinate mapping. Generation of the corroded surface and shared geometric mapping of the image and mask are performed within a single algorithmic pipeline, without constructing an explicit three-dimensional model or geometric mesh.

2.1 Overview of the Synthetic Data Generation Framework

First, a two-dimensional surface comprising a background texture, corrosion texture, and spatial corrosion pattern is generated together with a ground-truth mask. To prepare the inputs for cylindrical mapping, the dimensions of the two-dimensional surface and corresponding mask are adjusted to increase sampling density and reduce detail loss during projection.

A sediment texture layer is then added to the surface image. This layer is applied along both circumferential edges of the unwrapped surface, and corrosion covered by sediment is removed from the final mask of visible corrosion. The position of the virtual camera along the pipe axis is then set by simultaneously cropping the surface and mask from a specified longitudinal position.

Next, a non-uniform illumination field is applied to the cropped surface image to introduce brightness variations along the pipe. For each output image pixel, the corresponding coordinates on the unwrapped internal pipe surface are then computed using pixel-based cylindrical mapping. The resulting coordinate field is shared by the image and mask, so both undergo the same geometric transformation during projection.

After projection, the mask undergoes a central-region correction, while the image proceeds to the final scene-completion stage, which includes processing of the pipe-end region. The magnitude of this final processing effect depends on the illumination parameters and camera position. Finally, the synthetic image and its corresponding ground-truth mask are saved for segmentation. Figure 1 provides an overview of this processing pipeline.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

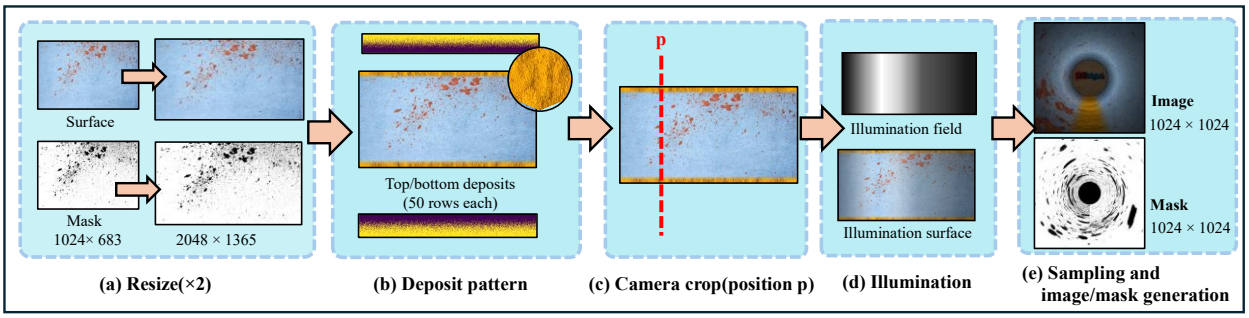


Figure 1. Overview of the generation of synthetic internal pipe corrosion images and corresponding masks, including two-dimensional surface preparation, shared cylindrical mapping, and separate final processing of the image and mask.

2.2 Generation of Two-Dimensional Corrosion Surfaces and Ground-Truth Masks

The first stage of the data generation framework creates a two-dimensional representation of the internal pipe surface before geometric transformations and imaging simulation are applied. Each synthetic surface consists of three main components: a background texture, a corrosion texture, and a binary pattern defining the corrosion region. Separating these components enables independent control over the background appearance, the texture of the corroded region, and the shape and spatial distribution of corrosion.

A collection of ice-surface textures with local heterogeneity and textural detail was used to generate the background. To increase visual diversity while preserving the spatial structure of the texture, the RGB channels were permuted and their intensities were scaled by different factors. If $T_{bg}^{(c)}(x,y)$ denotes the intensity of color channel c in the initial background texture, the transformation can be expressed as Equation (1):

$$\tilde{T}_{bg}^{(c)}(x,y) = \alpha_c T_{bg}^{(\pi(c))}(x,y), c \in \{R,G,B\} \quad (1)$$

where π(c) denotes the permutation of the RGB channels, α_c is the intensity scaling factor for channel c, and $\tilde{T}_{bg}^{(c)}$ is the background texture after color transformation. More suitable mappings were selected through visual review. Figure 2 shows examples of these transformations and the selected mappings.

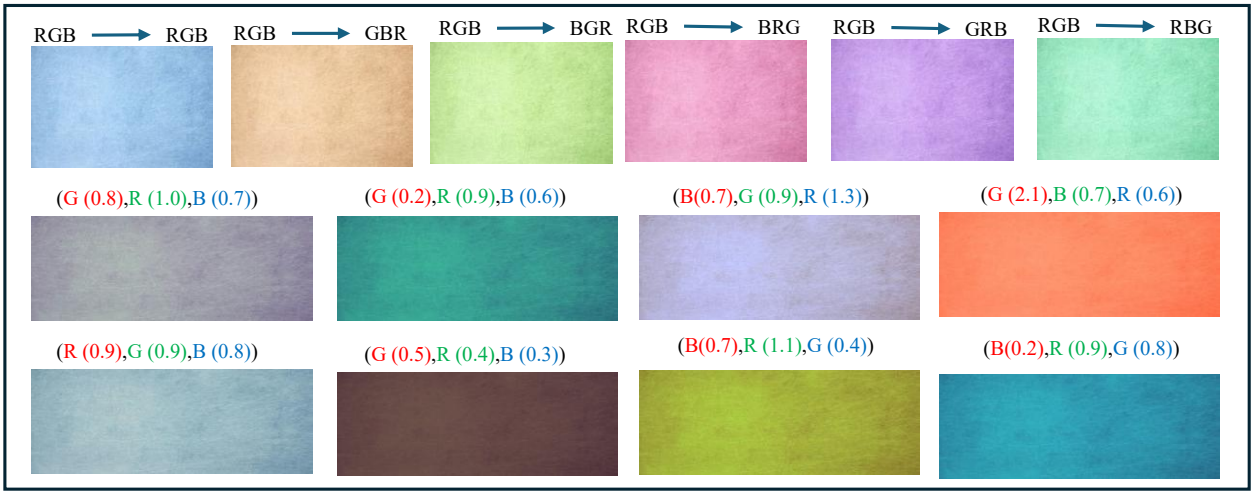


Figure 2. Background appearance diversification using RGB-channel permutation and channel-wise intensity scaling.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Independently of the background, a collection of corrosion textures was used to define the appearance of damaged regions, and a collection of irregular patterns was used to define the shape and spatial distribution of corrosion. After preparation, each spatial pattern was converted into a binary matrix B, where B(x,y) = 1 denotes corrosion and B(x,y) = 0 denotes the background. This separation defines the geometry and location of corrosion independently of its visual texture and enables different combinations of background, corrosion appearance, and spatial pattern. Figure 3 presents examples of the corrosion textures and binary patterns used.

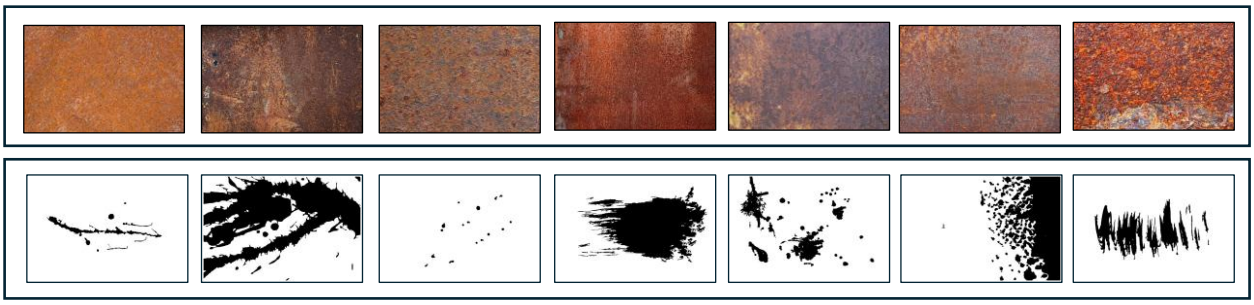


Figure 3. Representative corrosion textures and irregular binary patterns used to define corrosion appearance and spatial distribution.

After the input components are selected, the synthetic two-dimensional surface is generated using the binary mask to combine the corrosion texture and the modified background texture pixel by pixel:

$$I_{2D}(x,y) = B(x,y)T_{cor}(x,y) + [1 - B(x,y)]\tilde{T}_{bg}(x,y) \quad (2)$$

where I_{2D} is the resulting synthetic surface, T_{cor} is the corrosion texture, $\tilde{T}_{bg}$ is the background texture after color modification, and B is the binary corrosion mask. Thus, the pixel value is taken from the corrosion texture where B = 1 and from the background texture where B = 0.

Because the same matrix B directly specifies where the corrosion texture is applied, the two-dimensional ground-truth mask is obtained directly from that binary pattern:

$$M_{2D}(x,y) = B(x,y) \quad (3)$$

According to Equations (2) and (3), the synthetic image and its corresponding ground-truth mask are generated from a shared spatial definition, with no need for manual annotation at this stage. Figure 4 illustrates the combination of the input components and the simultaneous generation of the synthetic surface and corresponding mask.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

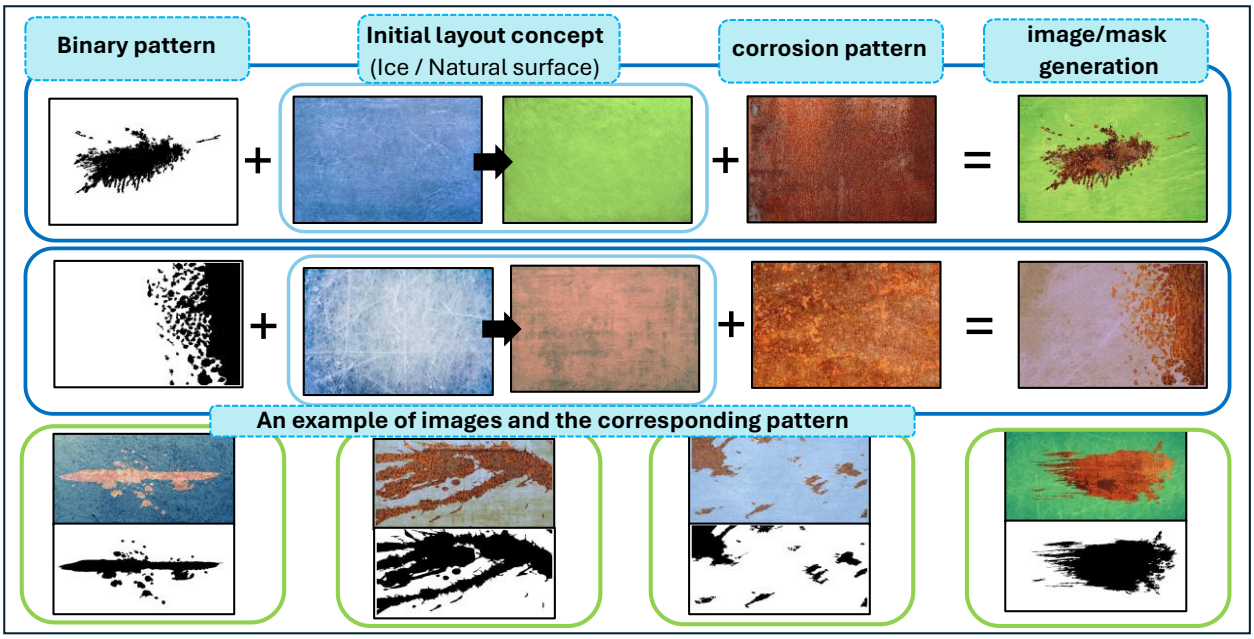


Figure 4. Generation of a two-dimensional synthetic corrosion surface and its corresponding reference mask from the background texture, corrosion texture, and binary corrosion pattern.

The synthetic surfaces generated at this stage had fixed dimensions of 1024 × 683 pixels and served as inputs to the subsequent mapping onto cylindrical pipe geometry.

2.3 Pixel-Based Cylindrical Mapping and Virtual-Camera Imaging

The two-dimensional surface and corresponding mask are resized from 1024 × 683 to 2048 × 1365 pixels to increase sampling density before projection and reduce detail loss caused by resampling. The corresponding coordinates on the unwrapped internal pipe surface are then computed directly in pixel space for each output pixel, without constructing an explicit three-dimensional model.

In the configuration used for data generation, the image produced by projection has dimensions of 1024 × 1024 pixels. Denoting the output image size by N, the center used in the implementation is defined by Equation (4):

$$c = \frac{N}{2} \tag{4}$$

In this study, N = 1024 gives c = 512. The radial distance of an output pixel (i,j) from the center is calculated using Equation (5):

$$\rho_{i,j} = \sqrt{(i-c)^2 + (j-c)^2} \tag{5}$$

In this coordinate system, the row index increases downward and the column index increases to the right.

To determine the circumferential position of each pixel on the internal pipe surface, the angle $\phi_{i,j}$ is calculated relative to the image center. A compact expression equivalent to the implementation is given by Equation (6):

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

$$\phi_{i,j} = \begin{cases} 180^{\circ} \ , & (i,j) = (c,c) \\ mod\left[\frac{180}{\pi}\ atan2(-(j-c), i-c), 360\ , \right. & \left. otherwise \right] \end{cases} \quad (6)$$

Under this convention, the bottom of the image corresponds to 0°, the left to 90°, the top to 180°, and the right to 270°. Consequently, the $0^{\circ}/360^{\circ}$ circumferential seam lies at the bottom of the pipe view.

To determine the longitudinal position corresponding to each pixel, the radial distance $\rho_{i,j}$ is mapped to an angular variable $\beta_{i,j}$ according to Equation (7):

$$\beta_{i,j} = 90^{\circ} - 30^{\circ}\frac{\rho_{i,j}}{512} \quad (7)$$

The virtual camera field of view was set to 120 degrees. In the mapping calculations, angles were measured relative to the central viewing axis; half of the field of view, equivalent to 60 degrees on either side of that axis, was therefore used.

The height H of the unwrapped surface is taken to represent one complete circumference of the internal pipe surface in pixel space. The equivalent cylinder radius is therefore obtained from Equation (8):

$$R_{px} = \frac{H}{2\pi} \quad (8)$$

In the current configuration, H = 1365 pixels, giving $R_{px} = 217\ px.$

This parameter is a geometric radius in pixel space and should not be interpreted as the physical pipe radius.

After $\phi_{i,j}$ and $\beta_{i,j}$ are calculated, the corresponding continuous coordinates on the unwrapped surface are obtained from Equations (9) and (10):

$$v^*_{i,j} = \frac{H}{360^{\circ}}\phi_{i,j} \quad (9)$$

$$u^*_{i,j} = R_{px}\ tan\left(\frac{\pi\beta_{i,j}}{180}\right) \quad (10)$$

Here, $v^*_{i,j}$ denotes the circumferential position on the unwrapped surface, and $u^*_{i,j}$ denotes the local longitudinal position.

The axial position of the virtual camera is applied before projection by cropping the surface. For position p, the surface and mask are taken from column p onward according to Equation (11):

$$T_p(v,u) = T(v,u+p) \quad (11)$$

The available local width is therefore

$$w_p = W_0 - p \quad (12)$$

where $W_0 = 2048$ pixels. Figure 5(b) schematically illustrates how the camera position and surface crop are applied, how the equivalent radius R_{px} is defined, and how the coordinates u^* and v^* are mapped.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

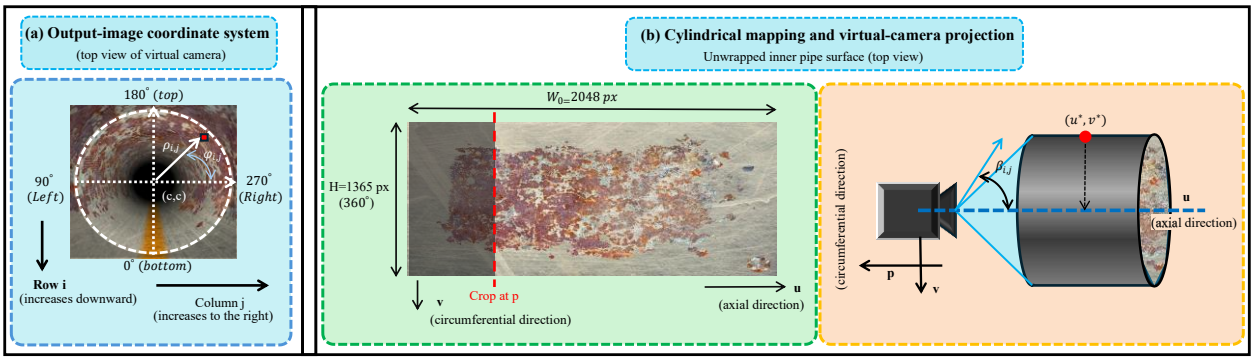


Figure 5. Schematic of pixel-based cylindrical mapping and virtual-camera imaging: (a) output image coordinate system and definitions of radial distance and circumferential angle; (b) application of camera position, axial mapping geometry, and coordinate transformation to the unwrapped internal pipe surface.

In the recorded configuration, the positions

$$p \in \{300,480,628,1024,1325,1400\}$$

were used to generate different views along the pipe. This parameter changes the longitudinal origin of the visible surface without altering the core projection equations.

During sampling, the continuous coordinates $u^*_{i,j}$ and $v^*_{i,j}$ are first rounded to the nearest integer indices, after which the boundary conditions are applied. If the rounded longitudinal coordinate exceeds the available width of the cropped surface, the pixel is considered out of bounds and the RGB value of the corresponding output image pixel is set to zero. Otherwise, the corresponding value is sampled from the two-dimensional surface. The same coordinate map is also used to project the ground-truth mask.

2.4 Modeling Appearance Conditions and Completing Scene Details

Three stages were added to the process to reduce visual artifacts: sediment modeling at the circumferential seam, non-uniform illumination, and terminal-region processing. Where a component covers part of the corrosion, the ground-truth mask is also modified to reflect the visible portion.

2.4.1 Sediment Modeling at the Circumferential Seam

The upper and lower edges of the unwrapped surface meet at the circumferential seam, and differences in their appearance can create an artificial boundary at the bottom of the pipe view. To reduce this effect, a sediment layer was added under the assumption that material accumulates at the bottom of the pipe.

To generate this layer, a granular texture is combined with an algorithmically generated transparency field. In the configuration used, the sediment region is defined as two 50-pixel strips along the upper and lower edges of the unwrapped surface. Transparency varies randomly along the surface length according to a weighted distribution, preventing the sediment boundary from having a completely uniform shape. The sediment texture is then blended with the surface image using this field.

Sediment covers part of the corrosion; the corresponding regions are therefore removed from the corrosion mask and assigned to the background so that the final labels describe visible corrosion.

Figure 6 presents an example of the sediment pattern and its effect on the image and corresponding mask.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

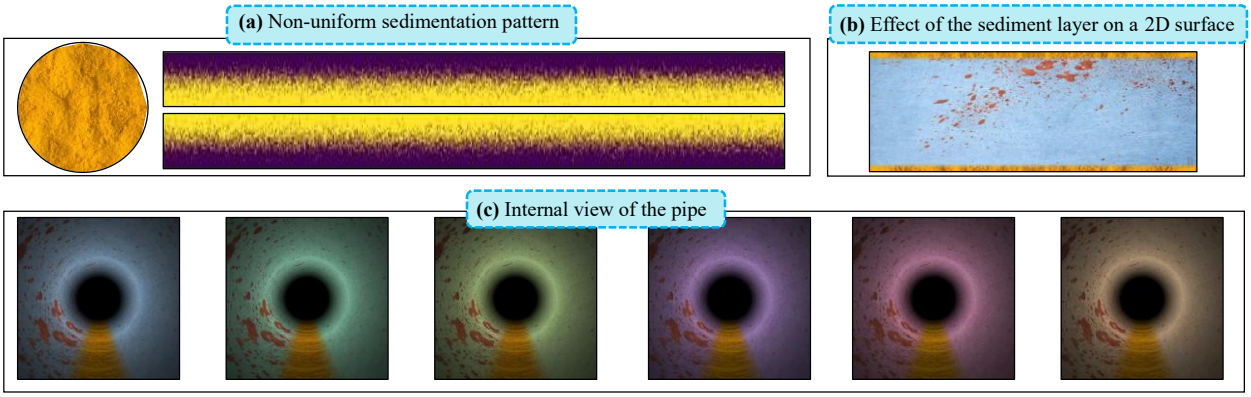


Figure 6. Sediment modeling at the circumferential seam of the surface: (a) an example of the non-uniform pattern used to define sediment distribution; (b) an example of the effect of applying the sediment layer to the two-dimensional surface; (c) the internal pipe view.

2.4.2 Non-Uniform Illumination Modeling

To establish the qualitative shape of the illumination field, the brightness distribution produced by a light source on a surface was examined in several configurations (Figure 7). This examination formed the basis for defining a one-dimensional illumination field.

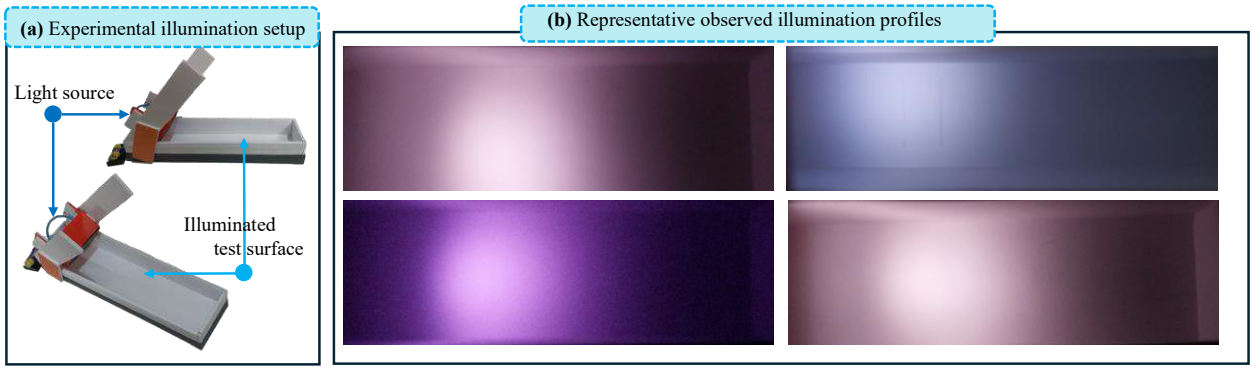


Figure 7. Experimental basis for the illumination model design: (a) the setup used for qualitative observation of light distribution on a surface; (b) examples of the illumination patterns observed during development.

In the final model, after the camera position is set and the surface is cropped, the illumination coefficient is constant within each column and varies along the longitudinal direction of the surface. This operation is applied to the image after cropping and before geometric projection.

For the surface widths used in the final configuration, the behavior of the implemented illumination field can be expressed as follows:

$$L(u) = 1 - \frac{(k + |u - u_0|)^{\gamma}}{\underset{0 \le q < W_p}{max}(k + |q - u_0|)^{\gamma}} \tag{13}$$

where u is the longitudinal surface coordinate after cropping, W_P is the cropped surface width, u_0 is the profile reference position, k is a shift term, and γ controls the shape of the intensity variation. In the configuration used, the parameters governing the illumination field are $u_0 = 400$, k = 100, and γ = 0.8.

Before projection, the illumination field obtained from Equation (13) is multiplied by the surface image according to Equation (14):

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

$$I_{ILL}(v,u,\ell) = I(v,u,\ell)L(u) \quad (14)$$

where ℓ denotes the color channel. This operation is applied only to the image; the ground-truth mask is not subjected to changes in illumination intensity.

Figure 8 presents examples of the profiles examined during development and the effect of the selected illumination model on the two-dimensional surface and internal pipe view.

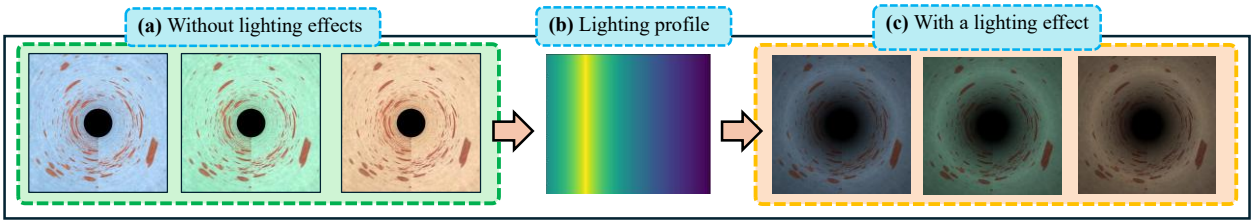


Figure 8. Implementation of the non-uniform illumination model in the image generator: (a) examples of internal pipe surface views without illumination; (b) the selected illumination profile; (c) examples of the effect of the selected illumination model on the internal pipe view.

2.4.3 Pipe-End Region Processing

In the initial projection, pixels near the viewing axis may map to longitudinal coordinates outside the bounds of the two-dimensional surface, producing a dark central region in the image. Additional processing of the pipe-end region is applied to the image after projection.

At this stage, the approximate dimensions of the central region are estimated from the projected image, and a supplementary texture is resized to match that region. The portion of the texture outside an approximately circular region is removed, and the remainder is blended with the image. The texture's contribution to this blend depends on a coefficient derived from the illumination model. The effect therefore varies with camera position and may be very limited or effectively zero in some views. Figure 9 compares examples before and after this processing stage.

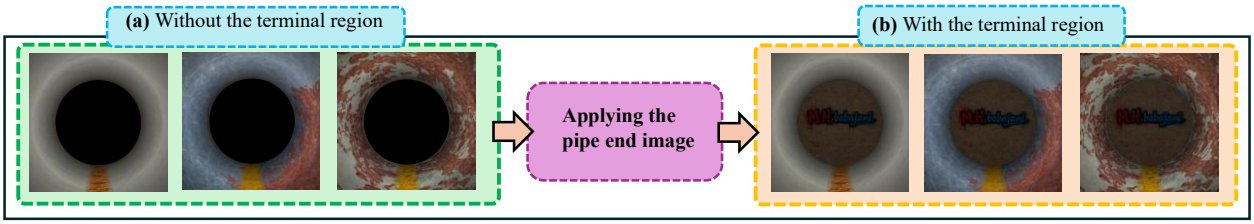


Figure 9. Development-stage examples of pipe-end region processing: (a) initial outputs after projection and (b) the corresponding examples after additional terminal-region processing.

2.5 Multi-View Generation by Varying the Virtual Camera Position

Varying the camera position while keeping the surface profile, including the background, corrosion distribution, and mask, fixed produces different views of the same underlying structure.

For each camera position, the two-dimensional surface and corresponding mask are cropped simultaneously from column p onward, and the number of columns remaining for projection is calculated using Equation (12).

In the configuration used, $W_0 = 2048$ pixels, and the six camera positions are defined as

$$p \in \{300,480,628,1024,1325,1400\}$$

Thus, each view samples a different portion of the same unwrapped surface, while the core cylindrical mapping equations remain unchanged.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Figure 10 schematically illustrates the placement of these discrete positions along the pipe axis and their effect on the usable surface region.

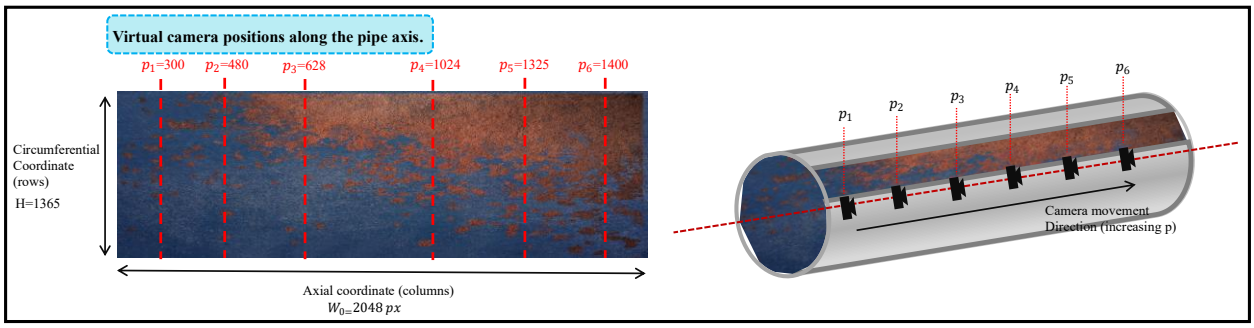


Figure 10. Schematic of the six virtual camera positions along the pipe axis and their corresponding positions on the unwrapped surface. As p increases, fewer columns remain available for mapping.

During initial generation, six discrete camera positions were considered for each surface profile, and one image–mask pair was generated at each position. Moving the camera changes the visible portion of the same surface, producing different views from a common profile. Figure 11 shows the six views generated from a fixed profile; the upper row contains the images, and the lower row contains the corresponding masks.

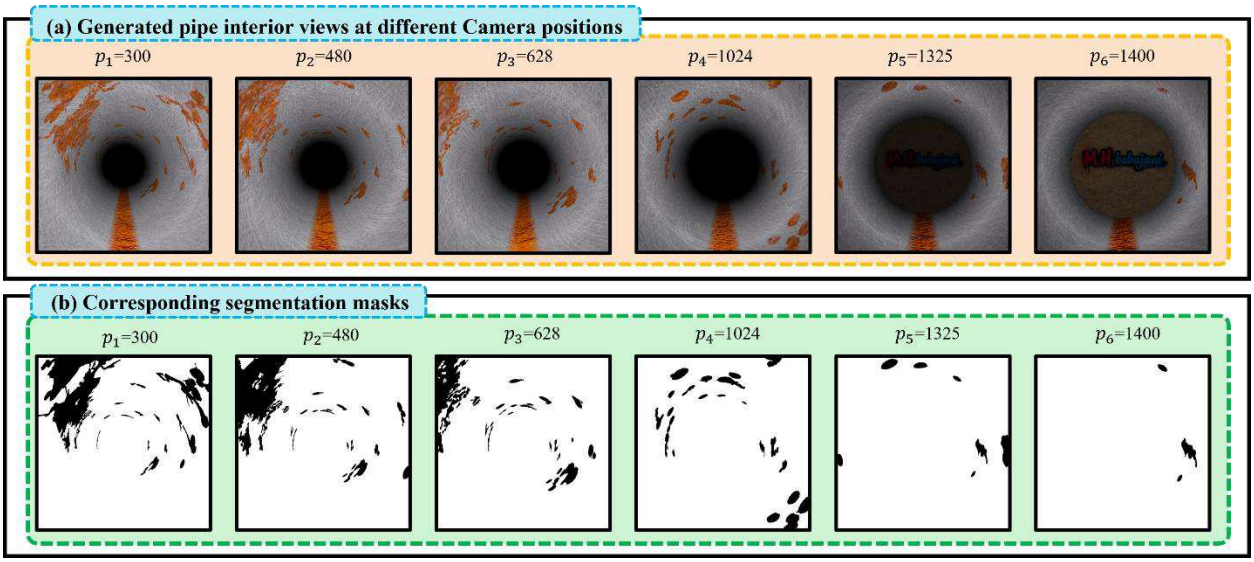


Figure 11. Six image–mask pairs generated from a fixed surface profile at six virtual camera positions: (a) internal pipe images and (b) corresponding segmentation masks.

3 Dataset Description

3.1 Dataset Composition

Initially, 1,244 two-dimensional corrosion surfaces were created, and six views of each surface produced a total of 7,464 image–mask pairs. During visual quality review, outputs that appeared excessively artificial or were visually or logically unsuitable were excluded. After 888 pairs were removed, the final dataset contained 6,576 image–mask pairs. Table 1 reports its composition, including 5,841 pairs containing corrosion and 735 pairs with empty masks.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Table 1. Composition of the final dataset and accounting of generated image–mask pairs.

Component	Count
Reconstructed profiles	1,244
Generated image–mask pairs before quality control	7,464
Generated image–mask pairs not retained in the final dataset	888
Retained image–mask pairs in the final dataset	6,576
Profiles with six retained views	1,022
Profiles with two retained views	222
Retained pairs with corrosion	5,841
Retained pairs with empty masks	735

3.2 Metadata and Sample Identification

For each sample, the metadata include the sample identifier, the filenames of the image and corresponding mask, the surface identifier, the view identifier, the camera position, and assignment to the training, validation, or test subset. The number and proportion of corrosion pixels, empty-mask status, and selected spatial properties of the corrosion region are also recorded at the sample level. The surface identifier is the source profile identifier and is used to group views generated from the same surface and to split the data at the profile level.

3.3 Dataset Splitting

The dataset was split at the profile level into training, validation, and test subsets using a target ratio of 70:15:15 and a fixed seed of 42.

Profile stratification for this split accounted for the number of retained views and the mean corrosion fraction of each profile. All views of a profile were assigned to the same subset, and no profile identifier was shared among the three subsets. Table 2 reports the numbers of profiles, pairs, and samples containing corrosion or empty masks.

Table 2. Composition of the training, validation and test subsets under profile-level splitting.

Subset	Profiles	Six-view profiles	Two-view profiles	Image-mask pairs	With corrosion	Empty masks
Training	872	716	156	4,608	4,086	522
Validation	186	153	33	984	866	118
Test	186	153	33	984	889	95
Total	1,244	1,022	222	6,576	5,841	735

3.4 File Formats and Mask Properties

Each sample consists of a JPEG image measuring 1024 × 1024 pixels and a corresponding PNG mask with the same dimensions and uint8 data type. The masks are encoded with two values: 0 for corrosion and 255 for the background.

The corrosion pixel fraction of each sample is calculated by dividing the number of zero-valued pixels by the total number of pixels in the mask. Across the dataset, this fraction has a mean of 10.70% and a median of 4.43%.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

3.5 Scope of the Final Dataset

The final dataset is a quality-controlled output of the generation framework. Varying the surface properties, corrosion pattern, sediment, illumination, and camera position within the implemented mechanisms enables additional image–mask pairs to be generated. Thus, the 6,576 image–mask pairs do not represent the generator's maximum capacity. Figure 12 illustrates this distinction.

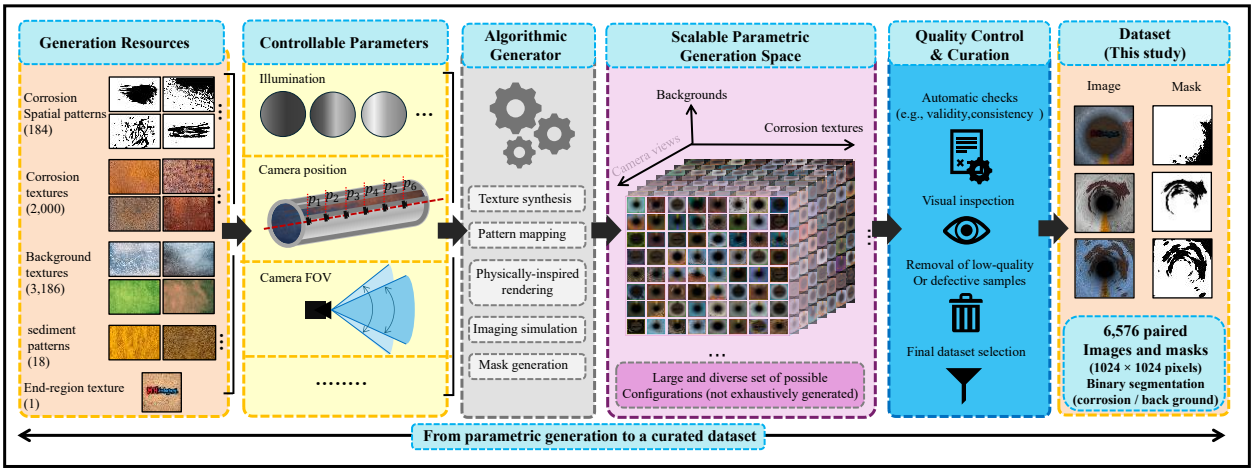


Figure 12. Conceptual overview of data generation, from the generator's resources and controllable parameters to formation of the final dataset after quality control.

4 Comparison with Related Datasets and Frameworks

To establish the context of the present study, related work is examined from two perspectives: how the data are acquired or generated, and the characteristics of the resulting datasets. Table 3 compares selected studies in terms of domain and data source, scale, learning task, ground-truth label type, geometry and camera model, and illumination or rendering approach.

Table 3. Comparison of selected datasets and data generation frameworks relevant to internal pipe corrosion segmentation.

Study	Domain / data source	Scale	Main task	Pixel-level GT	Internal-pipe / camera model	Illumination / rendering
K-Pipelines [8]	Synthetic pipeline-corrosion imagery	600 / 1,080 aug.	Classification	No	No explicit cylindrical/camera model	Prompt-based variation
Rodríguez Ospino et al. [17]	Physical DN600 pipe, prepared corrosion	1,200 original	Classification + segmentation	Yes, manual	Physical in-pipe camera	Controlled LED
Gao et al. [13]	Real drainage-pipe inspection	1,360	Segmentation	Yes, manual	Physical CCTV inspection	Not explicitly modeled
Wang et al. [18]	Real + GAN-generated steel corrosion	100 real + 400 virtual	Segmentation + severity recognition	Yes, semi-auto + manual refinement	No pipe geometry	Controlled photographic setup
BladeSynth [19]	Synthetic blade defects	12,500	Defect segmentation	Yes, automatic	3D virtual viewpoints	PBR lighting

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Study	Domain / data source	Scale	Main task	Pixel-level GT	Internal-pipe / camera model	Illumination / rendering
SYNOSIS [9]	Real + synthetic metal surfaces	5,400 synthetic + 270 real	Surface segmentation	Yes	Pinhole virtual camera	Modeled ring light
Yoo and Lee [14]	Synthetic internal pipe	50 perspective frames	IPM / reconstruction	No	Explicit cylindrical IPM + virtual camera	Not reported
Siu et al. [16]	Synthetic sewer-pipe defects + real CCTV images	1,296 synthetic images	Object detection	Semantic masks generated; used to derive bounding boxes	Explicit 3D pipe model + simulated CCTV viewpoints	Virtual lighting + style transfer
SORDI.ai [10]	Synthetic industrial scenes	1.19 M	Object detection	No, bounding boxes	Randomized virtual camera	PBR/path-traced
Proposed algorithmic framework and dataset	Synthetic internal-pipeline corrosion + laboratory image assessment	6,576 synthetic pairs (1,244 profiles) + 116 real images*	Binary corrosion segmentation + three-model benchmark	Yes, algorithmically generated (synthetic only)	Direct pixel-based cylindrical mapping + axial virtual camera (no 3D mesh)	Explicit non-uniform illumination + sediment modeling

*Note: The 116 real images from 10 physical pipe specimens were used for qualitative assessment only, without fine-tuning or pixel-level ground-truth masks.

Ramos et al. [8] introduced K-Pipelines, a collection of synthetic oil and gas pipeline corrosion images for binary classification. Annotations are provided at the image level, and no explicit model of cylindrical internal-surface geometry or camera projection is defined. By contrast, Rodríguez Ospino et al. [17] physically acquired images inside a DN600 steel pipe with prepared corrosion patterns on its internal surface and manually produced pixel-level masks for the corroded regions. Gao et al. [13] also used real drainage-network inspection images with pixel-level annotations for binary semantic segmentation. These two studies show that pixel-level segmentation of corrosion and other defects in internal pipe images has already been investigated; however, their data and ground-truth production depend on physical imaging and subsequent annotation.

In combining synthetic corrosion with pixel-level segmentation, Wang et al. [18] used real carbon-steel corrosion images together with virtual samples generated by StyleGAN3-t. The segmentation masks were produced through a semi-automatic SAM/ISAT-based process followed by manual refinement. Their images focus on flat steel surfaces, and the mask is not an inherent output of image synthesis.

BladeSynth [19] uses a three-dimensional rendering environment to create synthetic images of aero-engine blade surface defects, including corrosion, with automatically generated segmentation masks. SYNOSIS [9] also provides a procedural pipeline for metal surface inspection in which camera viewpoint, surface geometry, and illumination conditions are controlled during synthesis, producing synthetic images with pixel-level annotations. These frameworks are important examples of controllable image–mask pair generation, but neither was designed to model corrosion on the cylindrical internal surface of a pipe.

The geometry of internal pipe imaging was investigated separately by Yoo and Lee [14]. In terms of cylindrical geometry, perspective imaging, and changes in camera position, their study is among the closest

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

geometric precedents within the compared studies. Its objective, however, is neither corrosion data generation nor pixel-level ground truth for segmentation.

Siu et al. [16] also proposed a framework for generating synthetic images of internal sewer-pipe defects, using three-dimensional modeling, simulated imaging in a virtual environment, and image style transfer to increase training-data diversity. In their study, semantic masks generated in the simulation environment are used to derive defect bounding boxes, with the ultimate goal of improving object detection performance.

By contrast, the present framework generates internal pipe corrosion images and corresponding pixel-level masks for segmentation by directly mapping an unwrapped two-dimensional surface at the pixel level, without requiring explicit three-dimensional modeling. The distinction between the two approaches therefore lies in the data generation architecture, geometric representation, and type of labeled output, rather than simply in the use of synthetic images or a virtual camera.

With more than 1.19 million synthetic images across 111 object classes, SORDI.ai [10] demonstrates scalable data generation and variation in camera position. Its primary task, however, is object detection with two-dimensional bounding-box annotations, rather than pixel-level corrosion segmentation.

Based on the selected studies, the distinction of the present work lies in its specialized combination of components for internal pipe corrosion segmentation. Synthetic image generation and automated labeling are present in earlier studies [9, 16, 19], and cylindrical mapping has also been investigated previously [14]. The present contribution places direct mapping of the unwrapped surface at the center of a data generation pipeline for internal pipe corrosion segmentation.

Within this pipeline, the shared spatial definition of corrosion and identical coordinate field preserve image–mask correspondence during mapping. Unlike post-synthesis annotation [18], the initial mask is generated from the same spatial corrosion pattern. Sediment modeling, non-uniform illumination, and terminal-region processing also contribute to the visual representation of the pipe.

At the dataset level, recording profile identifiers and camera positions makes it possible to trace views from a common profile and perform profile-level splitting. The final dataset, derived from 1,244 profiles, includes samples containing corrosion and samples with empty masks. Compared with classification data [8] and datasets dependent on physical imaging and manual annotation [13, 17], its value lies in combining labels generated during synthesis, a traceable multi-view structure, and organization suited to segmentation evaluation.

5 Benchmark Experiments and Qualitative Assessment

To assess dataset utility, three representative segmentation architectures were tested in a controlled evaluation using synthetic images and their corresponding masks. The models were evaluated using common data, profile-level splits, and metrics within the reported training framework.

The benchmark used three models: U-Net [20] with a ResNet34 encoder [21], DeepLabV3+ [22] with a ResNet34 encoder, and SegFormer-B0 [23] with a MiT-B0 encoder. These architectures represent conventional CNN-based encoder–decoder methods, CNN-based methods with multiscale feature extraction, and Transformer-based architectures for segmentation, respectively. They were selected to assess the dataset's suitability for training and evaluating segmentation models.

5.1 Training Configuration and Evaluation Metrics

Table 4 summarizes the main benchmark training settings. All models were trained with input images measuring 1024 × 1024 pixels and weights pretrained on ImageNet [24]. A common training framework was used across architectures to examine dataset utility under the reported protocol.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Table 4. Training configuration of benchmark models

Parameter	Value
Input resolution	1024×1024
Encoder initialization	ImageNet pretrained
Optimizer	AdamW [25]
Initial learning rate	0.0001
Loss function	0.5 BCEWithLogits Loss + 0.5 Soft Dice Loss
Maximum training epochs	50
Batch size	Model dependent
Learning rate scheduler	ReduceLROnPlateau

The training loss combined Binary Cross Entropy with Logits Loss and Soft Dice Loss [26].

Model performance was evaluated using the Dice coefficient (F1-score), Intersection over Union (IoU), Precision, Recall, Specificity, and Matthews Correlation Coefficient (MCC). Given the predominance of background pixels, Pixel Accuracy may provide an overly optimistic picture of segmentation performance and was therefore not reported as a primary metric [27].

5.2 Quantitative Segmentation Results

For quantitative performance comparison, the three baseline architectures were evaluated on the held-out synthetic test set using common metrics. Table 5 presents the comparison across six segmentation metrics.

Table 5. Quantitative comparison of segmentation performance on the test set

Model	Dice/F1	IoU	Precision	Recall	Specificity	MCC
U-Net + ResNet34	0.9729	0.9471	0.9695	0.9763	0.9963	0.9695
DeepLabV3+ + ResNet34	0.9743	0.9499	0.9758	0.9729	0.9971	0.9712
SegFormer-B0 (MiT-B0)	0.9735	0.9484	0.9717	0.9754	0.9965	0.9703

On the synthetic test set, all three architectures achieved Dice scores above 0.97 and IoU values close to 0.95 (Table 5). These results demonstrate the dataset's utility for segmentation training and evaluation under the reported experimental conditions. DeepLabV3+ with a ResNet34 encoder achieved the highest Dice and IoU values, but its margins over the other two architectures were small and do not, by themselves, establish statistical superiority or generalizable performance.

5.3 Dataset-Oriented Performance Analysis

Model performance on corrosion-containing test images was examined from two perspectives: its relationship with corrosion coverage and the distribution of Dice scores across individual images. For the first analysis, each image's Corrosion Coverage Ratio was calculated by dividing the number of corrosion pixels in the ground-truth mask by the total number of image pixels. Thresholds for the four groups, Very small, Small, Medium, and Large, were determined from the distribution of this ratio among corrosion-containing training images and applied unchanged to group the corrosion-containing test images.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

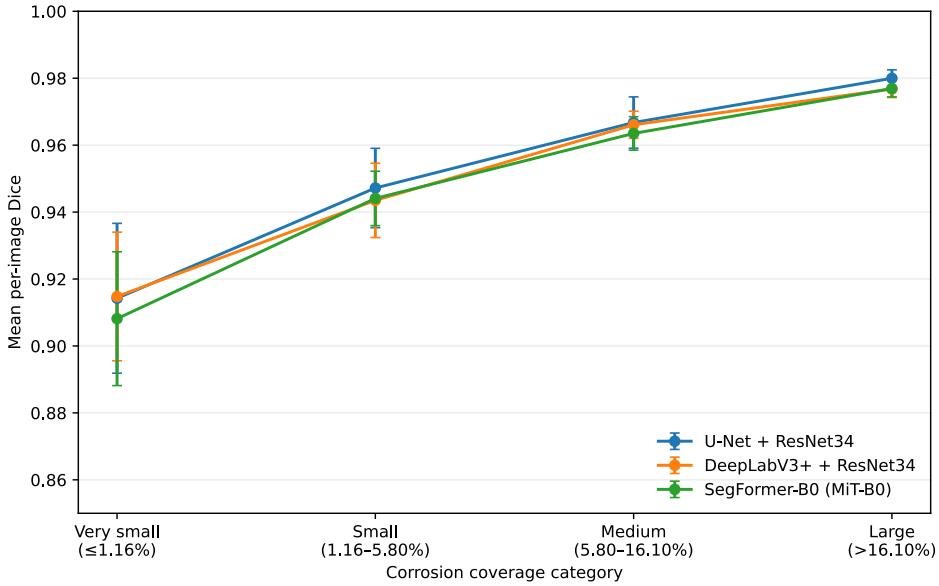


Figure 13. Segmentation performance analysis at different levels of corrosion coverage. Mean Dice scores are shown for the evaluated models across corrosion coverage groups.

As shown in Figure 13, samples with lower corrosion coverage generally had lower Dice scores than those with more extensive coverage, indicating a relationship between corrosion coverage and segmentation performance on the synthetic test set. To complement this analysis, the distribution of Dice scores across individual corrosion-containing test images was examined to characterize performance variability.

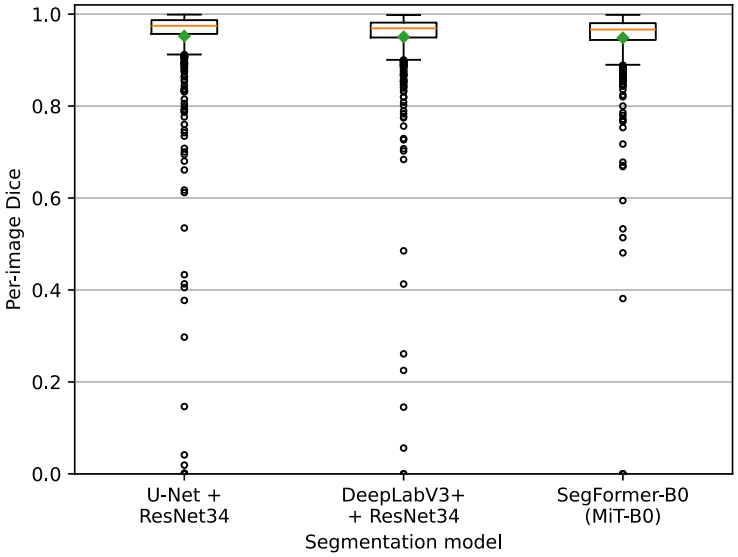


Figure 14. Distribution of Dice scores across individual test images for the evaluated segmentation models. The figure shows the spread of performance for the three models on corrosion-containing samples.

As shown in Figure 14, the Dice distributions of the three architectures are similar across corrosion-containing test images. Most samples have high Dice scores, although lower performance is observed for some images.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

5.4 Empty-Mask Sample Evaluation

In addition to corrosion-containing samples, model performance was examined on images without corrosion regions (empty-mask samples). The test set contained 95 corrosion-free samples, which were analyzed separately from the corrosion-containing samples. Table 6 presents the results.

Table 6. Performance evaluation of segmentation models on corrosion-free samples (empty-mask samples).

Model	Number of empty-mask samples	Images with false positives	Rate of images with false positives (%)	False-positive pixel ratio
U-Net + ResNet34	95	1	1.05	0.000007
DeepLabV3+ + ResNet34	95	0	0	0
SegFormer-B0 (MiT-B0)	95	3	3.16	0.000025

Among the 95 corrosion-free samples, the numbers of images with false-positive predictions were 1, 0, and 3 for U-Net, DeepLabV3+, and SegFormer-B0, respectively (Table 6). This analysis characterizes the behavior of the evaluated models when corrosion is absent within this test set.

5.5 Computational Characteristics

To complement the evaluation of dataset utility, the computational characteristics of the three baseline models were examined alongside their segmentation results. Table 7 reports parameter counts, training time, inference time and throughput, and GPU memory usage during training and testing.

Table 7. Computational characteristics of the benchmark models.

Model	Parameters (M)	Training time (h)	Training GPU memory (GB)	Inference time (ms/image)	Throughput (images/s)	Test GPU memory (GB)
U-Net + ResNet34	24.44	2.88	3.94	11.68	85.58	2.09
DeepLabV3+ + ResNet34	22.44	3.66	3.9	11.35	88.1	2.52
SegFormer-B0 (MiT-B0)	3.71	12.58	5.06	34.45	29.02	1.41

The results show that the evaluated models, which differ in architecture, have different computational requirements for training and inference.

5.6 Qualitative Analysis of Segmentation Predictions

To complement the quantitative evaluation, predictions from the three models were visually compared with ground-truth masks for selected synthetic test samples. Figure 15 presents examples of this comparison.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

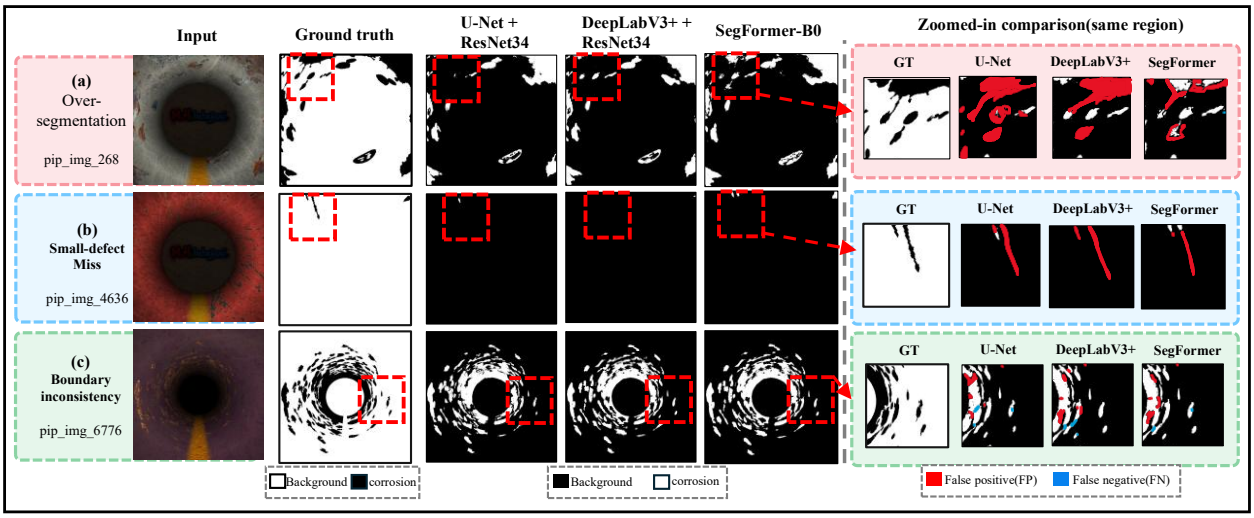


Figure 15. Qualitative segmentation examples showing the input image, ground-truth mask, and predictions from U-Net, DeepLabV3+, and SegFormer-B0.

As shown in Figure 15, samples with different corrosion shapes, sizes, and spatial distributions were selected for comparison. For these selected samples, the model outputs generally identify the corroded regions and show good overlap with the ground-truth masks, although discrepancies are visible at precise boundaries or in regions with limited coverage in some samples.

5.7 Supplementary Qualitative Assessment of Dataset Utility Using Laboratory Images

To complement the evaluation of dataset utility, a qualitative assessment was conducted using laboratory images of physical pipe specimens. This stage aimed to examine qualitatively how models trained on the synthetic dataset behave under imaging conditions outside the synthetic generation space.

The assessment used 116 RGB images from ten physical pipe specimens (PR-1 through PR-10). Images were standardized to 1024 × 1024 pixels for model input. U-Net, DeepLabV3+, and SegFormer-B0 were used with the weights obtained from training on the synthetic dataset, without retraining or fine-tuning on the laboratory images.

Selected examples are presented in Figure 16. Visual comparison of the outputs indicates that, for many samples, the models produced spatially coherent regions whose appearance was consistent with visible corrosion patterns. In a substantial proportion of the images, the models predicted similar overall locations and extents of corrosion. Under some more complex appearance conditions, however, their responses differed noticeably.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

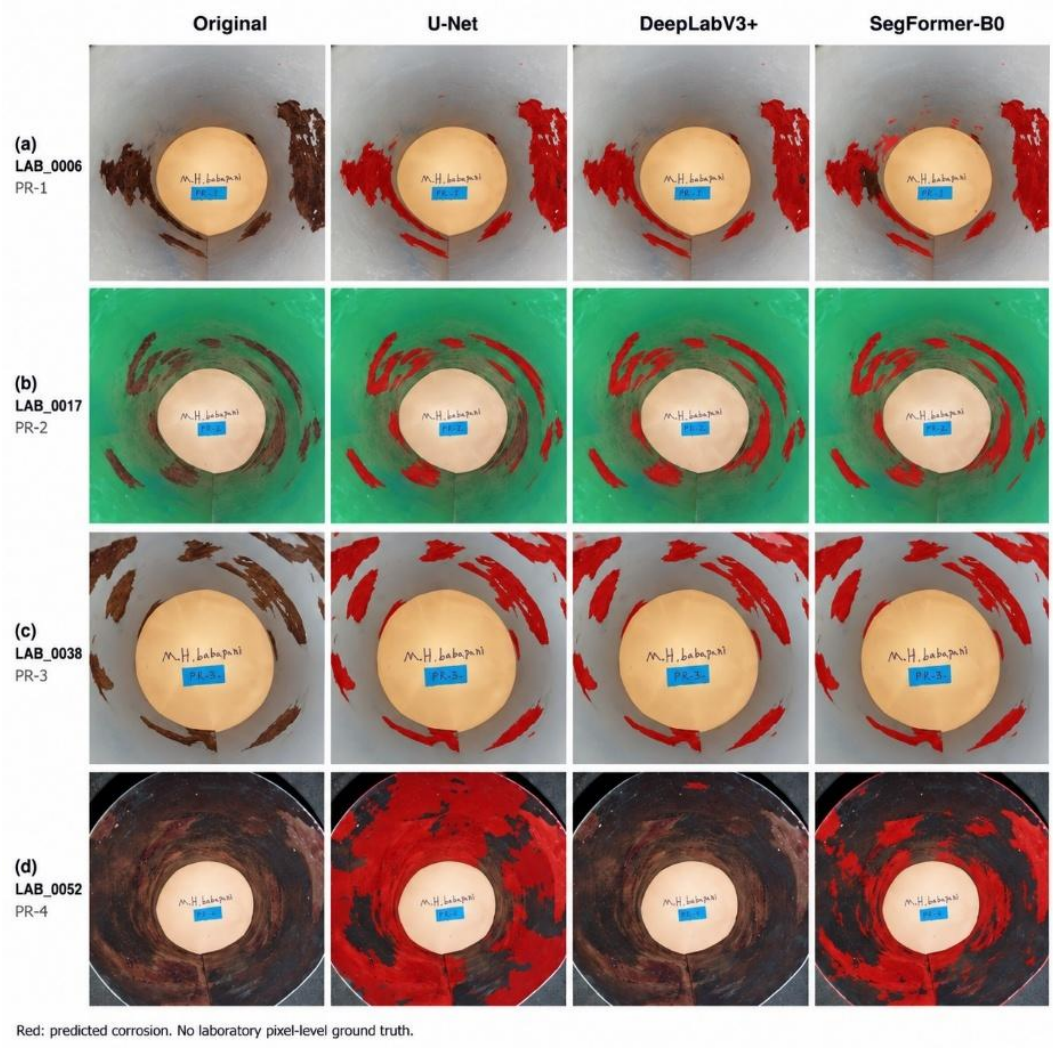


Figure 16. Qualitative comparison of U-Net, DeepLabV3+, and SegFormer-B0 predictions on selected laboratory images. Red regions indicate predicted corrosion.

Figure 17 presents four additional examples from PR-3, PR-5, and PR-7 to illustrate variation in appearance conditions and differences in the extent of the models' predictions.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

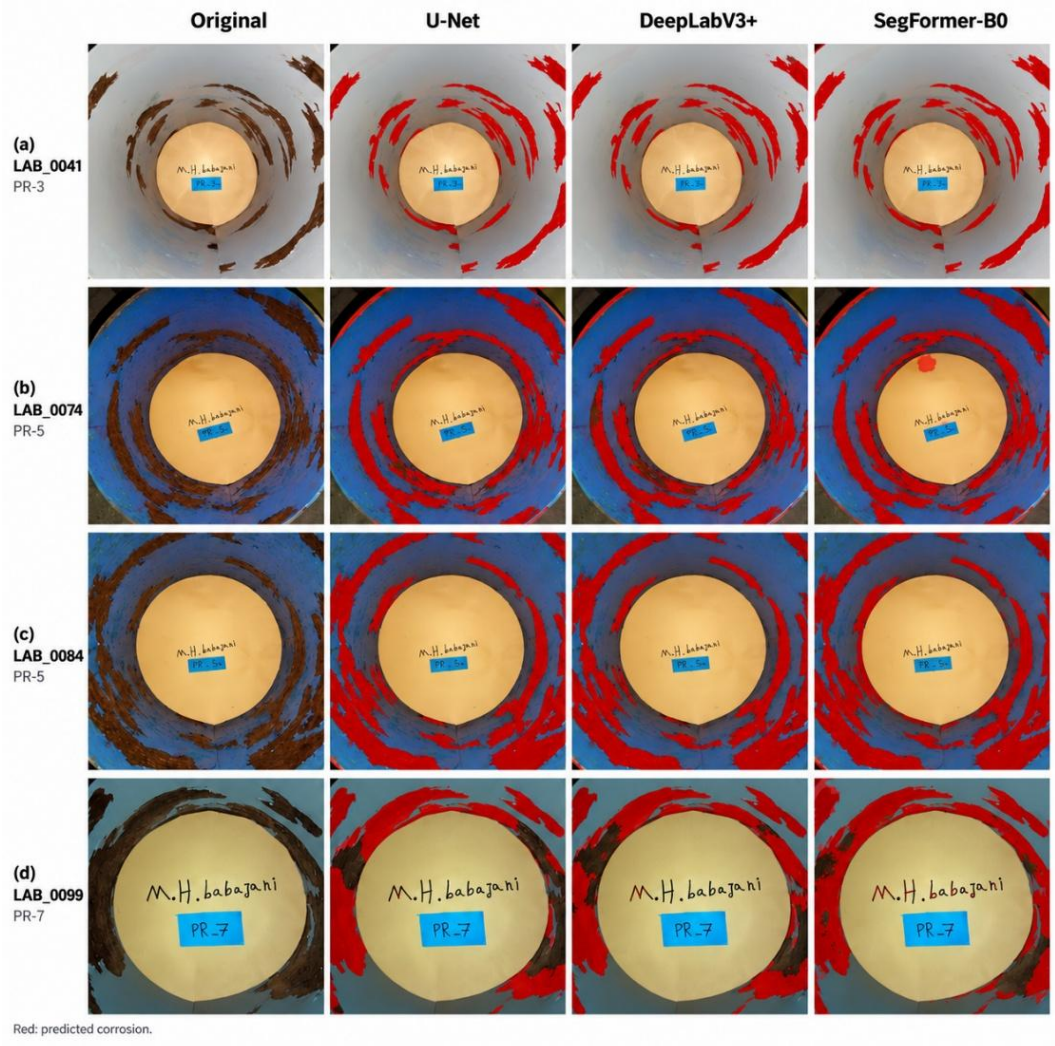


Figure 17. Supplementary comparison of model predictions on laboratory images under different appearance conditions. Red regions indicate predicted corrosion.

To describe differences among the three models' predictions on laboratory images, the fraction of pixels predicted as corrosion was compared for each image. In 26 of the 116 images, the difference between the highest and lowest predicted corrosion fractions across models exceeded 10% of the total image area. This measure indicates only differences between model outputs, not segmentation accuracy: without pixel-level ground-truth masks, Dice and IoU cannot be calculated and performance in the real-image domain cannot be quantitatively evaluated.

6 Discussion

6.1 Significance of Algorithmic and Dataset Contributions

The proposed framework integrates two-dimensional corrosion surface generation, direct pixel-based mapping to a cylindrical internal pipe view, and generation of corresponding masks within a single algorithmic pipeline. Without requiring an explicit three-dimensional model, it generates the image and initial mask from a shared spatial definition of corrosion, eliminating the need to annotate the synthetic views manually. Separating the spatial pattern from the visual texture also enables controlled variation.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

After quality control and metadata recording, the resulting dataset preserves the relationship between each view and its source profile. This structure enables profile-level splitting and evaluation on unseen profiles while preventing views of the same surface from being shared across subsets. The inclusion of both corrosion-containing and empty-mask samples also supports evaluation of segmentation and false-positive behavior.

6.2 Controllability of the Synthetic Data Generation Framework

Output diversity is obtained by adjusting the components defined in the synthesis process, in addition to introducing random variation. This controllability is restricted to the implemented mechanisms and ranges; it does not extend to independent control over every geometric, physical, and optical property of the inspection environment. Figure 18 shows the relationship between the components and outputs.

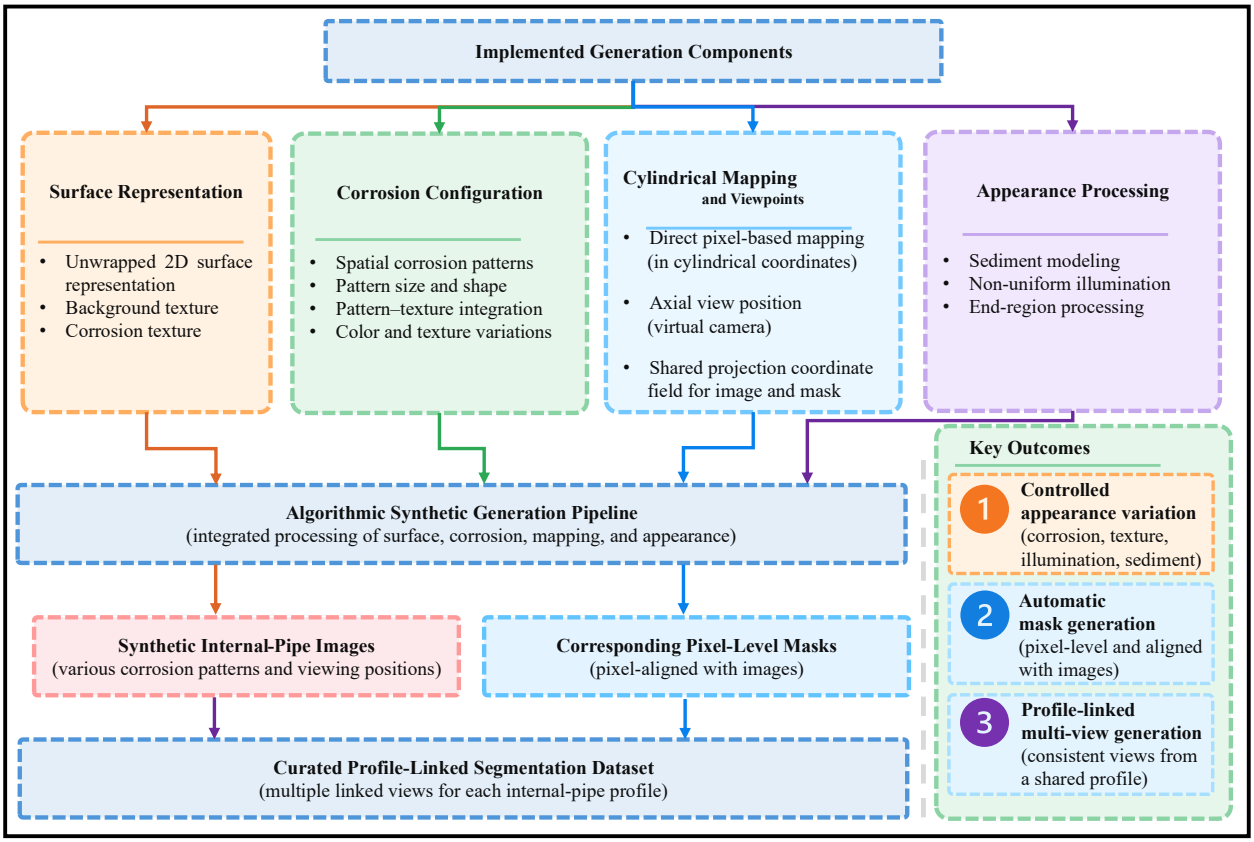


Figure 18. Schematic of the components implemented in the algorithmic framework for generating synthetic internal pipe corrosion data and their relationship to the production of pixel-level image–mask pairs and formation of a profile-linked multi-view dataset.

The ability to vary the generation components provides a basis for designing targeted experiments in future studies to examine segmentation performance on samples with different corrosion coverage, sizes, and spatial distributions.

6.3 Benchmark Validation and Dataset Utility

On the synthetic test set with profile-level splitting, all three architectures achieved Dice scores above 0.97. Analysis of corrosion coverage, image-level Dice distributions, and performance on corrosion-free samples also enabled model behavior to be examined beyond aggregate metrics. These results demonstrate the dataset's utility as a platform for training and controlled evaluation of corrosion segmentation in the synthetic domain.

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

Qualitative examination of real laboratory images showed that models trained on the synthetic dataset produced spatially coherent predictions for many samples without retraining. These observations complement the synthetic test results and provide preliminary evidence for investigating the dataset's utility with laboratory images.

6.4 Limitations and Domain Gap

Differences in surface texture, imaging noise, illumination, and corrosion morphology can create a gap between synthetic data and real inspection conditions [9, 12]. Variations in texture, sediment, illumination, and camera position within the framework represent some of these differences, but do not fully reproduce the physics and optics of an operational environment. Furthermore, shared mapping preserves the geometric correspondence between the image and mask; nevertheless, visually distinguishing labeled regions in the image may be difficult in very dark areas or for small corrosion patches.

The quantitative results are limited to the synthetic test set and the models examined. Assessment of laboratory images is also exclusively qualitative because pixel-level ground-truth masks are unavailable, and it cannot establish segmentation accuracy or generalization to operational conditions.

6.5 Future Directions

Future work could test combinations of synthetic and real data, transfer learning, domain adaptation, and synthetic–real training strategies to reduce the domain gap. Producing pixel-level ground-truth masks for real images would provide a basis for quantifying transfer performance. Extensions to the generation conditions should also be examined within specific inspection scenarios, with the effect of each change evaluated.

An independent study is underway to assess model performance and behavior on laboratory data more comprehensively, and its results will be reported separately.

Statements and Declarations

Ethics approval: Not applicable. This study did not involve human participants or animals. The dataset used is algorithmically generated and synthetic.

Consent to participate: Not applicable.

Consent for publication: Not applicable.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Funding: The authors did not receive support from any organization for the submitted work.

Data availability: The synthetic dataset generated and analyzed during the current study is available from the corresponding author on reasonable request.

Authors' contributions: [در اینجا می‌توانید سهم هر نویسنده را به طور مختصر بنویسید، مثلاً: S.N.M. conceived the study, designed the algorithm, and wrote the manuscript. A.F.S. and H.R.A. contributed to data generation and analysis. A.G. supervised the project and reviewed the final manuscript.]

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir

7 References

[1] Ali Hussein Khalaf; Ying Xiao; Ning Xu; Bohong Wu; Huan Li; Bing Lin; Zhen Nie; Junlei Tang. Emerging AI technologies for corrosion monitoring in oil and gas industry: A comprehensive review. Engineering Failure Analysis, 155, 107735 (2024). https://doi.org/10.1016/j.engfailanal.2023.107735

[2] Monika Rajendran; Deivalakshmi Subbian. Deep learning in corrosion assessment and control: a critical review of techniques and challenges. Corrosion Reviews, 44(1), 20240060 (2026). https://doi.org/10.1515/corrrev-2024-0060

[3] Md Mahadi Hasan Imran; Shahrizan Jamaludin; Ahmad Faisal Mohamad Ayob. A critical review of machine learning algorithms in maritime, offshore, and oil & gas corrosion research: A comprehensive analysis of ANN and RF models. Ocean Engineering, 295, 116796 (2024). https://doi.org/10.1016/j.oceaneng.2024.116796

[4] Subedi, A., Qian, C., Sadeghian, R., & Jahanshahi, M. R. (2025). CorFormer: a hybrid transformer-CNN architecture for corrosion segmentation on metallic surfaces. Machine Vision and Applications, 36(2), Article 45. https://doi.org/10.1007/s00138-025-01663-2

[5] Shahinuz Zaman; Po-Han Chen; Luh-Maan Chang. Real-Time Rust Segmentation on Steel Surface Using a Lightweight Transformer Deep Learning Model. Journal of Infrastructure Systems, 31(4), 04025033 (2025). https://doi.org/10.1061/JITSE4.ISENG-2747

[6] Shengyi Wang; Hai Nguyen; Rebekah Wilson; Brian Eick; Nora El-Gohary; Carolyn Ortiz. Deep CNN-based semi-supervised learning approach for identifying and segmenting corrosion in hydraulic steel and water resources infrastructure. Structural Health Monitoring, 24(4), 2229–2249 (2025). https://doi.org/10.1177/14759217241305039

[7] Pablo Malvido Fresnillo; Wael M. Mohammed; Saigopal Vasudevan; Jose A. Perez Garcia; Jose L. Martinez Lastra. Generation of realistic synthetic cable images to train deep learning segmentation models. Machine Vision and Applications, 35, 84 (2024). https://doi.org/10.1007/s00138-024-01562-y

[8] Leo Thomas Ramos; Edmundo Casas; Francklin Rivas-Echeverría. Synthetic generated data for intelligent corrosion classification in oil and gas pipelines. Intelligent Systems with Applications, 25, 200463 (2025). https://doi.org/10.1016/j.iswa.2024.200463

[9] Juraj Fulir; Natascha Jeziorski; Lovro Bosnar; Hans Hagen; Claudia Redenbach; Tobias Herrfurth; Marcus Trost; Thomas Gischkat; Petra Gospodnetić. SYNOSIS: Image Synthesis Pipeline for Machine Vision in Metal Surface Inspection. Sensors, 25(19), 6016 (2025). https://doi.org/10.3390/s25196016

[10] Chafic Abou Akar; Jimmy Tekli; Joe Khalil; Anthony Yaghi; Youssef Haddad; Abdallah Makhoul; Marc Kamradt. SORDI.ai: large-scale synthetic object recognition dataset generation for industries. Multimedia Tools and Applications, 84(17), 18263–18304 (2025). https://doi.org/10.1007/s11042-024-19731-6

[11] Christian Jung; Claudia Redenbach; Katja Schladitz. VoroCrack3d: An annotated semi-synthetic 3d image data set of cracked concrete. Data in Brief, 54, 110474 (2024). https://doi.org/10.1016/j.dib.2024.110474

[12] Katerina Kobrlova; Jiri Pavlas; Matous Cejnek; Milan Dvorak; Milan Ruzicka; Svatomir Slavik; Jan Vrba; Cyril Oswald. Segmentation-assisted corrosion classification on turboshaft blades using synthetic training data. npj Materials Degradation, 10, 83 (2026). https://doi.org/10.1038/s41529-026-00796-7

[13] Liang Gao; Xinxin Huang; Wanling Si; Feng Yang; Xu Qiao; Yaru Zhu; Tingyang Fu; Jianshe Zhao. An Improved HRNetV2-Based Semantic Segmentation Algorithm for Pipe Corrosion Detection in Smart City Drainage Networks. Journal of Imaging, 11(10), 325 (2025). https://doi.org/10.3390/jimaging11100325

[14] Sung Sic Yoo; Heung-Shik Lee. An Inverse Perspective Mapping-Based Approach for Generating Panoramic Images of Pipe Inner Surfaces. Sensors, 23(12), 5363 (2023). https://doi.org/10.3390/s23125363

[15] Johannes Künzel; Thomas Werner; Peter Eisert; Jan Waschnewski. Automatic Analysis of Sewer Pipes Based on Unrolled Monocular Fisheye Images. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019–2027 (2018). https://doi.org/10.1109/WACV.2018.00223

[16] ChunFai Siu; Mingzhu Wang; Jack C. P. Cheng. A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection. Automation in Construction, 137, 104213 (2022). https://doi.org/10.1016/j.autcon.2022.104213

*Corresponding author: Mohammad Hossein Babajani

Email addresses: moh.babajany@email.kntu.ac.ir

[17] Rafael Antonio Rodríguez Ospino; Cristhian Manuel Durán Acevedo; Jeniffer Katerine Carrillo Gómez. Detection of Uniform Corrosion in Steel Pipes Using a Mobile Artificial Vision System. Corrosion and Materials Degradation, 7(1), 21 (2026). https://doi.org/10.3390/cmd7010021

[18] Qiuji Wang; Haiyan Gong; Zhongheng Fu; Lingwei Ma; Yiran Li; Suyu Jin; Xiaogang Li; Dawei Zhang. High-precision corrosion degree nondestructive segmentation method with virtual and real synthetic data labeled by unsupervised learning. Computational Materials Science, 245, 113276 (2024). https://doi.org/10.1016/j.commatsci.2024.113276

[19] M. A. Mohammed Eltoum; Ehtesham Iqbal; Yahya Zweiri; Brain Moyo; Yusra Abdulrahman. BladeSynth: A High-Quality Rendering-Based Synthetic Dataset for Aero Engine Blade Defect Inspection. Scientific Data, 12, 1268 (2025). https://doi.org/10.1038/s41597-025-05563-y

[20] Olaf Ronneberger; Philipp Fischer; Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS 9351, 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28

[21] Kaiming He; Xiangyu Zhang; Shaoqing Ren; Jian Sun. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90

[22] Liang-Chieh Chen; Yukun Zhu; George Papandreou; Florian Schroff; Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. European Conference on Computer Vision (ECCV), 833–851 (2018). https://doi.org/10.1007/978-3-030-01234-2_49

[23] Enze Xie; Wenhai Wang; Zhiding Yu; Anima Anandkumar; Jose M. Alvarez; Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. Advances in Neural Information Processing Systems (NeurIPS), 34, 12077–12090 (2021). https://papers.nips.cc/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html

[24] Jia Deng; Wei Dong; Richard Socher; Li-Jia Li; Kai Li; Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848

[25] Ilya Loshchilov; Frank Hutter. Decoupled Weight Decay Regularization. International Conference on Learning Representations (ICLR) (2019). https://openreview.net/forum?id=Bkg6RiCqY7

[26] Fausto Milletari; Nassir Navab; Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. 2016 Fourth International Conference on 3D Vision (3DV), 565–571 (2016). https://doi.org/10.1109/3DV.2016.79

[27] Dominik Müller; Iñaki Soto-Rey; Frank Kramer. Towards a guideline for evaluation metrics in medical image segmentation. BMC Research Notes, 15, 210 (2022). https://doi.org/10.1186/s13104-022-06096-y

*Corresponding author: Mohammad Hossein Babajani
Email addresses: moh.babajany@email.kntu.ac.ir